CLAIMS

1. A method comprising:

processing an image using vector transformation followed by vector quantization to produce image values representative of the image;

forming a bitstream containing the image values; and

applying various protection levels to different parts of the bitstream.

2. A method as recited in claim 1, wherein the processing comprises:

computing a vector transformation of the image to produce multiple vectors;

calculating a mean value of each of the vectors;

quantizing the mean values to form quantized values;

removing the mean values from corresponding vectors to produce normalized vectors; and

coding the normalized vectors using vector quantization and a bit allocation map to produce vector indexes.

3. A method as recited in claim 1, wherein the vector transformation comprises:

sub-sampling the image into sub-images;

dividing each of the sub-images into blocks of pixels;

transforming each of the blocks of pixels to produce coefficients; and

regrouping the coefficients of the sub-images into one or more vectors.

4. A method as recited in claim 1, wherein the vector transformation forms vectors representative of the image, further comprising ordering the vectors in codebooks and assigning vector indexes to the vectors such that a bit error occurring at a less significant bit in a vector index results in less distortion than a bit error occurring at a more significant bit in the vector index.

5. A method as recited in claim 4, wherein the ordering comprises:

(a) selecting an initial vector;

(b) assigning a vector index to the initial vector;

(c) selecting a next vector that is nearest to the vector last assigned a vector index;

(d) assigning a next vector index to the next vector; and

(e) repeating the (c) selecting and the (d) assigning for all of the vectors.

6. A method as recited in claim 4, wherein the applying comprises assigning different channel codes to different parts of the bitstream.

7. A computer-readable medium comprising computer executable instructions that, when executed, direct a computing device to perform the method as recited in claim 1.

8. A method comprising:

ordering vectors representing a digital image in a codebook so that consecutive vectors are nearest neighbors to one another in n-dimensional space, where n>1;

assigning monotonically changing vector indexes to the vectors; and

storing the codebook containing the vectors and the vector indexes in memory.

9. A codebook data structure stored on a computer-readable medium that is produced as a result of the method of claim 8.

10. An image coding system, comprising:

a coder to process an image using vector transformation followed by vector quantization to produce image information, the coder forming a bitstream containing the image information; and

an error protection coder to apply various levels of protection to different parts of the bitstream.

11. An image coding system as recited in claim 10, wherein the coder is configured to compute the vector transformation by sub-sampling the image into sub-images, dividing each of the sub-images into blocks of pixels, transforming each of the blocks of pixels to produce coefficients, and regrouping the coefficients of the sub-images into one or more vectors.

12. An image coding system as recited in claim 10, wherein the vector transformation forms vectors representative of the image, the coder further comprising a codebook generator to generate an ordered codebook that contains the vectors and corresponding vector indexes that reference the vectors, the ordered codebook arranging the vectors such that a bit error occurring at a less significant bit in a vector index results in less distortion than a bit error occurring at a more significant bit.

13. An image coding system as recited in claim 12, wherein the codebook generator organizes the codebook by ordering the vectors such that a next vector is nearest to a previous vector and assigning monotonically changing vector indexes to the vectors.

14. An image coding system as recited in claim 10, wherein the error protection coder assigns different channel codes to different parts of the bitstream.

15. An image coding system as recited in claim 10, embodied as computer-readable instructions stored on a computer-readable medium.

16. An image distribution architecture, comprising:

a source having an image encoder to transform an image into multiple vectors and quantize the vectors to produce image values, the image encoder ordering the vectors in codebooks and assigning vector indexes to the vectors such that a bit error occurring at a less significant bit in a vector index results in less distortion than a bit error occurring at a more significant bit;

the source further having a UEP (Unequal Error Protection) coder that layers the vector indexes according to significance, the source transmitting a bit stream including at least the image values and the layered vector indexes over a communication channel; and

a destination receiving the bitstream, the destination having a decoder to recover the vectors using the vector indexes and reconstruct the image from the image values and the vectors.

17. An image distribution architecture as recited in claim 16, wherein the image encoder is configured to allocate different numbers of bits to different groups of the vectors.

18. An image distribution architecture as recited in claim 16, wherein the UEP coder assigns different channel codes to layers of the vector indexes.

19. A bitstream data structure stored on a computer-readable medium, comprising:

a first field containing mean values calculated from vectors produced from a digital image;

a second field containing a bit allocation map that allocates bits to various groups of the vectors; and

a third field containing vector indexes used to reference the vectors in a codebook, the vector indexes being layered.